

COS1004 Computer Systems Assignment 2

James Hassall

23 October 2019

Load Register

Syntax

```
ldr Rt, [Rn]
```

What it does?

ldr purpose is to move a value from memory to another register.

Example

```
ldr r8, [r10]
```

Special Case

```
LDRD Rt, Rt2, [Rn]
```

This is a special version of LDR in which the user moves a value that cannot be stored in a single register and stores it in two.

Move

Syntax

```
mov Rn, Decimal/Hex Value
```

What it does?

Mov copies the value given into the register specified

Example

```
mov r3, #100
```

Special Case

The mov command has a limit to the value that can be moved to the register using it, due to mov only working on 32 bit registers the maximum value can only be a total of 8 bits as the the 32 bits used are split up into the Opcode(20 bits), ROR(4bits) and finally the value(8 bits)

```
mov r2, #102100
```

this will end up flagging an error

OR

Syntax

```
orr rN, Decimal/Hex Value
```

What it does?

Orr can copy a value that is greater than an 8 bit value.

Example

```
orr r6,$000E
```

Special Case

The ORR instruction performs a bitwise OR operation on the values x and y.

Logical Shift Left

Syntax

```
lsl rN, Decimal/Hex Value
```

What it does?

LSL allows a value to be multiplied by the power 2, inserting bit positions and the value is shifted left.

Example

```
lsl r1,r4
```

Special Case

lsl can also be used with 3 commands where the rN is the destination register, the next is rM the register holding the first operand and the rS is the register holding the value to be shifted

```
lsl rN, rM, rS
```

Store Register

Syntax

```
str rN,[rM, Decimal/Hex Value]
```

What it does?

str wires the value into rN, from a reference pointer that hold an address(rM) and has the value to store in rN.

Example

```
str r1,[r0,#16]
```

Special Case

No special cases for store register

Branch

Syntax

```
Loop: - Label  
b Loop  
b - banch(goto)  
Loop - where to go to
```

What it does?

creates a loop at which the constantly runs.

Example

```
verticalloop:  
bls verticalloop
```

Special Case

mutliple conditions can be used to specify conditions such as ls which is ls than when a cmp function is called before the loop ends

Push

Syntax

```
push{rN}
```

What it does?

push adds the current values onto the stack

Example

```
push{r1,r5,r2}
```

Special Case

only when popping items from the stack they have to be in the exact same order on the same line.

Pop

Syntax

```
pop{rN}
```

What it does?

pop removes the current values from the stack

Example

```
pop{r1,r5,r2}
```

Special Case

Refer to push documentation

Bit Clear

Syntax

```
bic rN,rM,Decimal/Hex Value
```

What it does?

clears the bits inside the register if they are set in bit mask

Example

```
bic r1,r1,#1024
```

Special Case

There a few condition flags for the command S updates the destination register and source pointer

Test Bit

Syntax

```
tst rN,Decimal/Hex Value
```

What it does?

updates the APSR register and bitmask allowing the value to be tested against

Example

```
tst r0,#1024
```

Special Case

No special cases for test bit

Add

Syntax

```
add rN, rM
```

What it does?

adds a value to an existing register

Example

```
add r1, #5
```

Sub

Syntax

```
sub rN, rM
```

What it does?

subtracts a value to an existing register

Example

```
sub r1, #5
```

Compare

Syntax

```
cmp rN, rM
```

What it does?

compares a value/register to a register

Example

```
cmp r1, #5
```

Special Case

Comparing against a value is faster than comparing against a register

Logical Shift Right

Syntax

```
lsr rN, Decimal/Hex Value
```

What it does?

LSr allows a value to be multiplied by the power 2, inserting bit positions and the value is shifted right.

Example

```
lsr r1,r4
```

Special Case

lsr can also be used with 3 commands where the rN is the destination register, the next is rM the register holding the first operand and the rS is the register holding the value to be shifted

```
lsr rN, rM, rS
```

Multiply

Syntax

```
mul rD, Rn, Rm
```

What it does?

multiplies two registers together and inserts into a new register

Example

```
mul r1,r4,r7
```

Special Case

mul can also be used to multiply into the same register by specifying the first r1, the second to r1 and the final to the register you are multiplying

```
mul r1, r1, r6
```


Test Equivalence

Syntax

```
teq rN,Operand2
```

What it does?

Teq tests the equivalence of the register against either a value or another register.

Example

```
teq r0,r5
```

Special Case

TEQEQ is an Exclusive OR operation on the two values this is useful for testing the sign of a value

Load Register Exclusive

Syntax

```
ldrex rT,[rN]
```

What it does?

Loads a values into memory that has an exclusive tag to the physical address of the current processor, and clears any other exclusivity tags as well

Example

```
ldrex r0,[r5]
```

Special Case

LDREX is specifically used in conjunction with STREX in order to keep the performances its advised to keep the instruction in between these operations to a minimum.

Store Register Exclusive

Syntax

```
strex rD,rT,[rN]
```

What it does?

STREX performs a conditional store. this means that it remeans once it is performed it will clear any outstanding exclusivity tags meaning any other function can access it

Example

```
strex r0,r1,[r5]
```

Special Case

Refer to LDREX for Special Case

Logical Exclusive OR

Syntax

```
eor rD,rN,Operand
```

What it does?

EOR performs an exclusive OR operation on the values.

Example

```
strex r0,r1,#52
```

Rotate Right

Syntax

```
ror rD, rM, 0perand
```

What it does?

Stores a value that has been rotated by another value/register into a new register.

Example

```
ror r0,r1,r7
```